

Peer-Review 2: UML

Manta, Selva, Simoni

Gruppo 15

7 maggio 2022

Valutazione del diagramma UML delle classi del gruppo 25.

1 Lati positivi

- L'utilizzo degli handler evita l'utilizzo estensivo di switch per decifrare il messaggio ricevuto e rende più semplice e leggibile il codice
- Avere un controller dedicato ai turni è sicuramente una buona scelta per non avere un unico controller molto grande e più prone a errori/dimenticanze
- Il modo in cui sono gestite le classi dei messaggi permette una certa modularità ed estendibilità dell'applicazione
- Una view virtuale aiuta sicuramente come layer di intermezzo, per focalizzarci tutta la struttura principale di comunicazione. Così risulta anche più semplice un eventuale cambio di interfaccia dei messaggi, che non richiederebbe una modifica del model/controller

2 Lati negativi

- Nonostante il commento positivo sugli Handle, non è chiaro dove venga eseguito il relativo overload e dove vengano dichiarati i metodi handle(xyz). Ci viene quindi il dubbio che in realtà non vengano sfruttati correttamente e venga comunque usato una sorta di switch nella lettura del messaggio

- Il model ha ancora diversi problemi segnalati nella review precedente
- Gli observer sono nella teoria una buona scelta, ma in pratica (in quanto deprecati) andrebbero sostituiti con strumenti più recenti e aggiornati come i listeners
- Sempre in merito ad Observer e Observable, sono state definite una classe e una interfaccia aventi lo stesso nome. Ciò porta a conflitti con le omonime classi di libreria `java.util.Observer` e `java.util.Observable`
- L'enumerazione `MessageType` potrebbe essere ridondante: ogni messaggio ha già una propria sottoclasse distintiva che permette di riconoscerlo senza usare un identificatore ulteriore

3 Confronto tra le architetture

Anche la nostra architettura prevede l'utilizzo di handler e di messaggi con relative sottoclassi specializzate, quindi l'approccio è stato molto simile. Riteniamo però che la nostra scelta di usare dei Listeners invece degli Observers possa essere preferibile visto che gli ultimi sono ormai deprecati. Tutto sommato il modello generale sembra ben ideato e costruito, anche se presenta comunque delle pecche già segnalate nella vecchia review. Sicuramente risolvendole si potrebbe aspirare a un progetto molto solido e flessibile.